EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	989	703/14.ccls.	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2006/05/20 16:41
L2	402	(model adj check$3) and @ad<"20010113"	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2006/05/20 17:14
L3	50	L2 and trace	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2006/05/20 17:17
L4	20	L2 and disjoint	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2006/05/20 17:18
L5	3	(disjoint adj trace$1) and @ad<"20010113"	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2006/05/20 17:21
L6	421	(multiple adj trace$1) and @ad<"20010113"	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2006/05/20 17:21

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1	(mutually-disjoint or (mutually adj disjoint) and trace$1 and model).clm.	US-PGPUB	OR	OFF	2006/05/20 18:03
L2	1	(mutually-disjoint or (mutually adj disjoint) and trace$1).clm.	US-PGPUB	OR	OFF	2006/05/20 18:03
L3	5	(mutually-disjoint or (mutually adj disjoint)).clm.	US-PGPUB	OR	OFF	2006/05/20 18:03
L4	3	((reachable adj set$1) and trace$1 and intersection$1).clm.	US-PGPUB	OR	OFF	2006/05/20 18:08
L5	3	(disjoint and intersection$1 and trace$1 and intersection$1).clm.	US-PGPUB	OR	OFF	2006/05/20 18:08

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2	"6691078".pn.	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2006/05/20 18:11
L2	136	(disjoint near path$1) and @ad<"20010113"	US-PGPUB; USPAT; EPO; DERWENT	OR	OFF	2006/05/20 18:12

Results for "((model check*<and>multiple trace*)) <and> (pyr >= 1951 <and> pyr <= 2001)"
Your search matched **1** of **1351118** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

☐ 1. **Static analysis and dynamic steering of time-dependent systems**
Vicario, E.;
Software Engineering, IEEE Transactions on
Volume 27, Issue 8, Aug. 2001 Page(s):728 - 748
Digital Object Identifier 10.1109/32.940727

AbstractPlus | References | Full Text: PDF(1496 KB) **IEEE JNL**
Rights and Permissions

Results for "((model check*<and>disjoint trace*)) <and> (pyr >= 1951 <and> pyr <= 2001)"
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

e-mail printer friendly

» Search Options

View Session History

New Search

Modify Search

((model check*<and>disjoint trace*)) <and> (pyr >= 1951 <and> pyr <= 2001) Search

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistance revising your search.

Results for "((model check*<and>trace*)<and>disjoint) <and> (pyr >= 1951 <and> pyr <..."
Your search matched **43** of **1351118** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

e-mail printer friendly

» Search Options

View Session History

New Search

Modify Search

((model check*<and>trace*)<and>disjoint) <and> (pyr >= 1951 <and> pyr <= 2001) Search

Check to search only within this results set

Display Format: Citation Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

view selected items Select All Deselect All

1. **Handling obstacles in goal-oriented requirements engineering**
van Lamsweerde, A.; Letier, E.;
Software Engineering, IEEE Transactions on
Volume 26, Issue 10, Oct. 2000 Page(s):978 - 1005
Digital Object Identifier 10.1109/32.879820
AbstractPlus | References | Full Text: PDF(976 KB) IEEE JNL
Rights and Permissions

2. **Symbolic model checking for sequential circuit verification**
Burch, J.R.; Clarke, E.M.; Long, D.E.; McMillan, K.L.; Dill, D.L.;
Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on
Volume 13, Issue 4, April 1994 Page(s):401 - 424
Digital Object Identifier 10.1109/43.275352
AbstractPlus | Full Text: PDF(2324 KB) IEEE JNL
Rights and Permissions

3. **FunState-an internal design representation for codesign**
Strehl, K.; Thiele, L.; Gries, M.; Ziegenbein, D.; Ernst, R.; Teich, J.;
Very Large Scale Integration (VLSI) Systems, IEEE Transactions on
Volume 9, Issue 4, Aug. 2001 Page(s):524 - 544
Digital Object Identifier 10.1109/92.931229
AbstractPlus | References | Full Text: PDF(448 KB) IEEE JNL
Rights and Permissions

4. **A logical characterization of bisimulation for labeled Markov processes**
Desharnais, J.; Edalat, A.; Panangaden, P.;
Logic in Computer Science, 1998. Proceedings. Thirteenth Annual IEEE Symposium on
21-24 June 1998 Page(s):478 - 487
Digital Object Identifier 10.1109/LICS.1998.705681
AbstractPlus | Full Text: PDF(152 KB) IEEE CNF
Rights and Permissions

5. **Managing conflicts in goal-driven requirements engineering**
van Lamsweerde, A.; Darimont, R.; Letier, E.;
Software Engineering, IEEE Transactions on
Volume 24, Issue 11, Nov. 1998 Page(s):908 - 926
Digital Object Identifier 10.1109/32.730542
AbstractPlus | References | Full Text: PDF(448 KB) IEEE JNL
Rights and Permissions

6. **Mechanizing CSP trace theory in higher order logic**
Camilleri, A.J.;
Software Engineering, IEEE Transactions on
Volume 16, Issue 9, Sept. 1990 Page(s):993 - 1004

Digital Object Identifier 10.1109/32.58786

AbstractPlus | Full Text: PDF(1084 KB) **IEEE JNL**
Rights and Permissions

7. **Supervisory control of a rapid thermal multiprocessor**
Balemi, S.; Hoffmann, G.J.; Gyugyi, P.; Wong-Toi, H.; Franklin, G.F.;
Automatic Control, IEEE Transactions on
Volume 38, Issue 7, July 1993 Page(s):1040 - 1059
Digital Object Identifier 10.1109/9.231459

AbstractPlus | Full Text: PDF(1964 KB) **IEEE JNL**
Rights and Permissions

8. **Where do operations come from? A multiparadigm specification technique**
Zave, P.; Jackson, M.;
Software Engineering, IEEE Transactions on
Volume 22, Issue 7, July 1996 Page(s):508 - 528
Digital Object Identifier 10.1109/32.538607

AbstractPlus | References | Full Text: PDF(1980 KB) **IEEE JNL**
Rights and Permissions

9. **Logics for hybrid systems**
Davoren, J.M.; Nerode, A.;
Proceedings of the IEEE
Volume 88, Issue 7, July 2000 Page(s):985 - 1010
Digital Object Identifier 10.1109/5.871305

AbstractPlus | References | Full Text: PDF(1344 KB) **IEEE JNL**
Rights and Permissions

10. **How to make apples from oranges in UML**
Selonen, P.; Koskimies, K.; Sakkinen, M.;
System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on
Jan 3-6 2001 Page(s):10 pp.

AbstractPlus | Full Text: PDF(192 KB) **IEEE CNF**
Rights and Permissions

11. **Good enough versus high assurance software testing and analysis methods**
Howden, W.E.;
High-Assurance Systems Engineering Symposium, 1998. Proceedings. Third IEEE International
13-14 Nov. 1998 Page(s):166 - 175
Digital Object Identifier 10.1109/HASE.1998.731609

AbstractPlus | Full Text: PDF(2100 KB) **IEEE CNF**
Rights and Permissions

12. **Strand spaces: why is a security protocol correct?**
Fabrega, F.J.T.; Herzog, J.C.; Guttman, J.D.;
Security and Privacy, 1998. Proceedings. 1998 IEEE Symposium on
3-6 May 1998 Page(s):160 - 171
Digital Object Identifier 10.1109/SECPRI.1998.674832

AbstractPlus | Full Text: PDF(140 KB) **IEEE CNF**
Rights and Permissions

13. **Early experience with the Visual Programmer's WorkBench**
Rubin, R.V.; Walker, J., II; Golin, E.J.;
Software Engineering, IEEE Transactions on
Volume 16, Issue 10, Oct. 1990 Page(s):1107 - 1121
Digital Object Identifier 10.1109/32.60292

AbstractPlus | Full Text: PDF(1272 KB) **IEEE JNL**
Rights and Permissions

14. **Analysis of real-time rule-based systems with behavioral constraint assertions specified in Estella**
Cheng, A.M.K.; Browne, J.C.; Mok, A.K.; Rwo-Hsi Wang;
Software Engineering, IEEE Transactions on
Volume 19, Issue 9, Sept. 1993 Page(s):863 - 885
Digital Object Identifier 10.1109/32.241770

AbstractPlus | Full Text: PDF(1716 KB) IEEE JNL
Rights and Permissions

15. **Reliable software and communication. I. An overview**
Chung, F.R.K.;
Selected Areas in Communications, IEEE Journal on
Volume 12, Issue 1, Jan. 1994 Page(s):23 - 32
Digital Object Identifier 10.1109/49.265700
AbstractPlus | Full Text: PDF(1088 KB) IEEE JNL
Rights and Permissions

16. **Timing analysis of Ada tasking programs**
Corbett, J.C.;
Software Engineering, IEEE Transactions on
Volume 22, Issue 7, July 1996 Page(s):461 - 483
Digital Object Identifier 10.1109/32.538604
AbstractPlus | References | Full Text: PDF(2196 KB) IEEE JNL
Rights and Permissions

17. **Principles and methods of testing finite state machines-a survey**
Lee, D.; Yannakakis, M.;
Proceedings of the IEEE
Volume 84, Issue 8, Aug. 1996 Page(s):1090 - 1123
Digital Object Identifier 10.1109/5.533956
AbstractPlus | References | Full Text: PDF(3560 KB) IEEE JNL
Rights and Permissions

18. **Completeness and consistency in hierarchical state-based requirements**
Heimdahl, M.P.E.; Leveson, N.G.;
Software Engineering, IEEE Transactions on
Volume 22, Issue 6, June 1996 Page(s):363 - 377
Digital Object Identifier 10.1109/32.508311
AbstractPlus | References | Full Text: PDF(1472 KB) IEEE JNL
Rights and Permissions

19. **Automatic symbolic verification of embedded systems**
Alur, R.; Henzinger, T.A.; Pei-Hsin Ho;
Software Engineering, IEEE Transactions on
Volume 22, Issue 3, March 1996 Page(s):181 - 201
Digital Object Identifier 10.1109/32.489079
AbstractPlus | References | Full Text: PDF(2112 KB) IEEE JNL
Rights and Permissions

20. **The Compositional Security Checker: a tool for the verification of information flow security properties**
Focardi, R.; Gorrieri, R.;
Software Engineering, IEEE Transactions on
Volume 23, Issue 9, Sept. 1997 Page(s):550 - 571
Digital Object Identifier 10.1109/32.629493
AbstractPlus | References | Full Text: PDF(372 KB) IEEE JNL
Rights and Permissions

21. **Use of sequencing constraints for specification-based testing of concurrent programs**
Carver, R.H.; Kuo-Chung Tai;
Software Engineering, IEEE Transactions on
Volume 24, Issue 6, June 1998 Page(s):471 - 490
Digital Object Identifier 10.1109/32.689403
AbstractPlus | References | Full Text: PDF(144 KB) IEEE JNL
Rights and Permissions

22. **Experiences using lightweight formal methods for requirements modeling**
Easterbrook, S.; Lutz, R.; Covington, R.; Kelly, J.; Ampo, Y.; Hamilton, D.;
Software Engineering, IEEE Transactions on
Volume 24, Issue 1, Jan. 1998 Page(s):4 - 14
Digital Object Identifier 10.1109/32.663994

AbstractPlus | References | Full Text: PDF(520 KB) **IEEE JNL**
Rights and Permissions

23. **Supervisory hybrid systems**
Lemmon, M.D.; He, K.X.; Markovsky, I.;
Control Systems Magazine, IEEE
Volume 19, Issue 4, Aug. 1999 Page(s):42 - 55
Digital Object Identifier 10.1109/37.777788

AbstractPlus | Full Text: PDF(2652 KB) **IEEE JNL**
Rights and Permissions

24. **Modeling and formal verification of the Fairisle ATM switch fabric using MDGs**
Tahar, S.; Xiaoyu Song; Cerny, E.; Zijian Zhou; Langevin, M.; Ait-Mohamed, O.;
Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on
Volume 18, Issue 7, July 1999 Page(s):956 - 972
Digital Object Identifier 10.1109/43.771178

AbstractPlus | References | Full Text: PDF(268 KB) **IEEE JNL**
Rights and Permissions

25. **Hierarchical finite state machines with multiple concurrency models**
Girault, A.; Bilung Lee; Lee, E.A.;
Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on
Volume 18, Issue 6, June 1999 Page(s):742 - 760
Digital Object Identifier 10.1109/43.766725

AbstractPlus | References | Full Text: PDF(312 KB) **IEEE JNL**
Rights and Permissions

Published before February 2001
Terms used **model checker trace**

Found **126** of **117,446**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 126 Result page: **1** 2 3 4 5 6 7 next

Relevance scale

1 A practical method for verifying event-driven software

Gerard J. Holzmann, Margaret H. Smith

May 1999 **Proceedings of the 21st international conference on Software engineering**

Publisher: IEEE Computer Society Press

Full text available: pdf(1.40 MB) Additional Information: full citation, references, citings, index terms

Keywords: case studies, feature interactive, formal methods, model checking, reactive systems, software testing, software verification, telephone call processing

2 Using model checking to generate tests from requirements specifications

Angelo Gargantini, Constance Heitmeyer

October 1999 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 7th European software engineering conference held jointly with the 7th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-7**, Volume 24 Issue 6

Publisher: Springer-Verlag, ACM Press

Full text available: pdf(1.44 MB) Additional Information: full citation, abstract, references, citings, index terms

Recently, many formal methods, such as the SCR (Software Cost Reduction) requirements method, have been proposed for improving the quality of software specifications. Although improved specifications are valuable, the ultimate objective of software development is to produce software that satisfies its requirements. To evaluate the correctness of a software implementation, one can apply black-box testing to determine whether the implementation, given a sequence of system inputs, produces the ...

3 Bandera: extracting finite-state models from Java source code

James C. Corbett, Matthew B. Dwyer, John Hatcliff, Shawn Laubach, Corina S. Păsăreanu, Robby, Hongjun Zheng

June 2000 **Proceedings of the 22nd international conference on Software engineering**

Publisher: ACM Press

Full text available: pdf(345.15 KB) Additional Information: full citation, abstract, references, citings, index terms

Finite-state verification techniques, such as model checking, have shown promise as a cost-effective means for finding defects in hardware designs. To date, the application of these techniques to software has been hindered by several obstacles. Chief among these is the problem of constructing a finite-state model that approximates the executable behavior of the software system of interest. Current best-practice involves hand-construction of models which is expensive (prohibitive for all but ...

Keywords: abstract interpretation, model checking, model extraction, program specialization, program verification, slicing

4 Three approximation techniques for ASTRAL symbolic model checking of infinite state real-time systems

Zhe Dang, Richard A. Kemmerer

June 2000 **Proceedings of the 22nd international conference on Software engineering**

Publisher: ACM Press

Full text available: pdf(359.05 KB) Additional Information: full citation, abstract, references, citings, index terms

ASTRAL is a high-level formal specification language for real-time systems. It has structuring mechanisms that allow one to build modularized specifications of complex real-time systems with layering. Based upon the ASTRAL symbolic model checler reported in [13], three approximation techniques to speed-up the model checking process for use in debugging a specification are presented. The techniques are random walk, partial image and dynamic environment generation. Ten mutation tests on a rai ...

Keywords: ASTRAL, formal methods, formal specification and verification, model checking, real-time systems, state machines, timing requirements

5 HSIS: a BDD-based environment for formal verification

A. Aziz, F. Balarin, S.-T. Cheng, R. Hojati, T. Kam, S. C. Krishnan, R. K. Ranjan, T. R. Shiple, V. Singhal, S. Tasiran, H.-Y. Wang, R. K. Brayton, A. L. Sangiovanni-Vincentelli

June 1994 **Proceedings of the 31st annual conference on Design automation**

Publisher: ACM Press

Full text available: pdf(91.11 KB) Additional Information: full citation, references, citings, index terms

6 Verification of time partitioning in the DEOS scheduler kernel

John Penix, Willem Visser, Eric Engstrom, Aaron Larson, Nicholas Weininger

June 2000 **Proceedings of the 22nd international conference on Software engineering**

Publisher: ACM Press

Full text available: pdf(111.58 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes an experiment to use the Spin model checking system to support automated verification of time partitioning in the Honeywell DEOS real-time scheduling kernel. The goal of the experiment was to investigate whether model checking could be used to find a subtle implementation error that was originally discovered and fixed during the standard formal review process. To conduct the experiment, a core slice of the DEOS scheduling kernel was first translated without abstraction ...

7 Decoupling synchronization from local control for efficient symbolic model checking of statecharts

William Chan, Richard J. Anderson, Paul Beame, David H. Jones, David Notkin, William E. Warner

May 1999 **Proceedings of the 21st international conference on Software engineering**

Publisher: IEEE Computer Society Press

Full text available: pdf(1.31 MB) Additional Information: full citation, references, citings, index terms

Keywords: binary decision diagrams, fault tolerance, formal methods, formal verification, software specification, statecharts, symbolic model checking

8 Fault origin adjudication

Karthikeyan Bhargavan, Carl A. Gunter, Davor Obradovic

August 2000 **Proceedings of the third workshop on Formal methods in software practice**

Publisher: ACM Press

Full text available: pdf(522.20 KB) Additional Information: full citation, abstract, references, citings, index terms

When a program P fails to satisfy a requirement R supposedly ensured by a detailed specification S that was used to implement P, there is a question about whether the problem arises in S or in P. We call this determination fault origin adjudication and illustrate its significance in various software engineering contexts. The primary contribution of this paper is a fra ...

9 Formal verification of FIRE: a case study

Jae-Young Jang, Shaz Qadeer, Matt Kaufmann, Carl Pixley

June 1997 **Proceedings of the 34th annual conference on Design automation DAC '97**

Publisher: ACM Press

Full text available: pdf(93.19 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

We present our experiences with the formal verification of an automotivechip used to control the safety features in a car. We useda BDD based model checker in our work. We describe our verificationmethodology for verifying a very complicated property on arelatively large design. We also describe the bugs that were foundand present our views on how to make model checking an effectiveintegrated part of the design flow for complex hardware systems.

10 Using the ASTRAL model checker to analyze mobile IP

Zhe Dang, Richard A. Kemmerer

May 1999 **Proceedings of the 21st international conference on Software engineering**

Publisher: IEEE Computer Society Press

Full text available: pdf(1.16 MB) Additional Information: full citation, references, citings, index terms

Keywords: ASTRAL, Encryption protocols, formal methods, formal specification and verification, real-time systems, state machines, timing requirements

11 Forward model checking techniques oriented to buggy designs

Hiroaki Iwashita, Tsuneo Nakata

November 1997 **Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design**

Publisher: IEEE Computer Society

Full text available: pdf(91.22 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Forward model checking is an efficient symbolic model checking method for verifying realistic properties of sequential circuits and protocols. In this paper, we present the techniques that modify the order of state traversal on forward model checking, and that dramatically improve average CPU time for finding design errors. A failing property has to be checked again and again to analyze the bug until it is corrected. The techniques, therefore, can have significant impacts on actual verification ...

Keywords: formal verification, symbolic state traversal, symbolic model checking, forward model checking

12 Alcoa: the alloy constraint analyzer

Daniel Jackson, Ian Schechter, Hya Shlyahter

June 2000 **Proceedings of the 22nd international conference on Software**

engineering
Publisher: ACM Press

Full text available: pdf(145.10 KB) Additional Information: full citation, abstract, references, citings, index terms

Alcoa is a tool for analyzing object models. It has a range of uses. At one end, it can act as a support tool for object model diagrams, checking for consistency of multiplicities and generating sample snapshots. At the other end, it embodies a lightweight formal method in which subtle properties of behaviour can be investigated.Alcoa's input language, Alloy, is a new notation based on Z. Its development was motivated by the need for a notation that is more closely tailored to ob ...

Keywords: constraint satisfaction, formal specifications, model checking, object models, relational logic, software analysis

13 Fitting formal methods into the design cycle

K. L. McMillan
June 1994 **Proceedings of the 31st annual conference on Design automation**
Publisher: ACM Press
Full text available: pdf(325.61 KB) Additional Information: full citation, references, citings, index terms

14 A formal basis for architectural connection

Robert Allen, David Garlan
July 1997 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 6 Issue 3
Publisher: ACM Press

Full text available: pdf(463.23 KB) Additional Information: full citation, abstract, references, citings, index terms, review

As software systems become more complex, the overall system structure—or software architecture—becomes a central design problem. An important step toward an engineering discipline of software is a formal basis for describing and analyzing these designs. In the article we present a formal approach to one aspect of architectural design: the interactions among components. The key idea is to define architectural connectors as explicit semantic entities. These are specified as a col ...

Keywords: WRIGHT, formal models, model-checking, module interconnection, software analysis

15 Speeding up symbolic model checking by accelerating dynamic variable reordering

Christoph Meinel, Christian Stangier
March 2000 **Proceedings of the 10th Great Lakes symposium on VLSI**
Publisher: ACM Press

Full text available: pdf(500.80 KB) Additional Information: full citation, abstract, references, citings, index terms

Symbolic Model checking is a widely used technique in sequential verification. As the size of the OBDDs and also the computation time depends on the order of the input variables, the verification may only succeed if a well suited variable order is chosen. Since the characteristics of the represented functions are changing, the variable order has to be adapted dynamically. Unfortunately, dynamic reordering strategies are often very time consuming and sometimes do not provide any improvement of ...

16 Safety critical systems based on formal models

Lars Asplund, Kristina Lundqvist
December 2000 **ACM SIGAda Ada Letters**, Volume XX Issue 4
Publisher: ACM Press
Full text available: pdf(732.05 KB) Additional Information: full citation, abstract, index terms

The Ravenscar profile for high integrity systems using Ada 95 is well defined in all real-

time aspects. The complexity of the run-time system has been reduced to allow full utilization of formal methods for applications using the Ravenscar profile. In the Mana project a tool set is being developed including a formal model of a Ravenscar compliant run-time system, a gnat compatible run-time system, and an ASIS based tool to allow for the verification of a system including both COTS and code that ...

17 Verification techniques for cache coherence protocols

Fong Pong, Michel Dubois
March 1997 **ACM Computing Surveys (CSUR)**, Volume 29 Issue 1
Publisher: ACM Press

Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, references, citings, index terms

In this article we present a comprehensive survey of various approaches for the verification of cache coherence protocols based on state enumeration, (symbolic model checking, and symbolic state models. Since these techniques search the state space of the protocol exhaustively, the amount of memory required to manipulate that state information and the verification time grow very fast with the number of processors and the complexity of the protocol mechanism ...

Keywords: cache coherence, finite state machine, protocol verification, shared-memory multiprocessors, state representation and expansion

18 Efficient generation of counterexamples and witnesses in symbolic model checking

E. M. Clarke, O. Grumberg, K. L. McMillan, X. Zhao
January 1995 **Proceedings of the 32nd ACM/IEEE conference on Design automation**
Publisher: ACM Press
Full text available: pdf(225.22 KB) Additional Information: full citation, references, citings, index terms

19 Strategic directions in real-time and embedded systems

John A. Stankovic
December 1996 **ACM Computing Surveys (CSUR)**, Volume 28 Issue 4
Publisher: ACM Press
Full text available: pdf(209.23 KB) Additional Information: full citation, references, citings, index terms

20 Formal specification and verification of a dataflow processor array

Thomas A. Henzinger, Xiaojun Liu, Shaz Qadeer, Sriram K. Rajamani
November 1999 **Proceedings of the 1999 IEEE/ACM international conference on Computer-aided design**
Publisher: IEEE Press

Full text available: pdf(98.54 KB) Additional Information: full citation, abstract, references, citings, index terms

We describe the formal specification and verification of the VGI parallel DSP chip [1], which contains 64 compute processors with ~30K gates in each processor. Our effort coincided in time with the "informal" verification stage of the chip. By interacting with the designers, we produced an abstract but executable specification of the design which embodies the programmer's view of the system. Given the size of the design, an automatic check that even one of the 64 processors sati ...

Scholar

Results **1 - 10** of about **212** for **"model checker" disjoint trace**. (**0.15** seconds)

All articles Recent articles

NUSMV: a new symbolic **model checker** - group of 12 »
AJ Cimatti, EJ Clarke, FJ Giunchiglia, MJ Roveri - International Journal on Software Tools for Technology ..., 2000 - Springer
... N U SMV: a new symbolic **model checker** ... NuSMV is the result of the reengineering and reim- plementation of the CMU SMV [26,47] symbolic **model checker**. ...
Cited by 108 - Web Search

[PS] Automated software testing using model-checking - group of 3 »
J Callahan, F Schneider, S Easterbrook - Proceedings 1996 SPIN Workshop, 1996 - cis.upenn.edu
... While the partitions created by a CCC are **disjoint** ... specific tests into one or more partitions using a **model checker**. We can determine if a **trace** belongs to a ...
Cited by 38 - View as HTML - Web Search

... Verification of a Behavioural Subset of UML Statechart Diagrams Using the SPIN **Model-checker** - group of 6 »
DA Latella, IA Majzik, MA Massink - Formal Aspects of Computing, 1999 - Springer
... as the work constitutes a basis for a PROMELA/SPIN based **model-checker** for UML ... where F is a finite set of sequential automata with mutually **disjoint** sets of ...
Cited by 110 - Web Search - BL Direct

A partial approach to model checking - group of 8 »
P Godefroid, P Wolper - Logic in Computer Science, 1991. LICS'91., Proceedings of ..., 1991 - ieeexplore.ieee.org
... i (where S, is the set of states of A,) are pairwise **disjoint**. words accepted by AG (all states of AG considered ac- cepting). We define the **trace** behavior of ...
Cited by 159 - Web Search - BL Direct

[PS] Is your **model checker** on time - group of 7 »
L Aceto, F Laroussinie - Proc. 24th Int. Symp. Math. Found. Comp. Sci.(MFCS'99), ..., 1999 - lsv.ens-cachan.fr
... Is your **Model Checker** on Time? ... The complexity of implementation verication for (concurrent) programs is studied in, eg, [39,56,63], where both **trace**- and tree ...
Cited by 29 - View as HTML - Web Search - BL Direct

Using a **model checker** to test safety properties - group of 5 »
P Ammann, W Ding, D Xu - International Conference on Engineering of Complex Computer ..., 2001 - doi.ieeecs.org
... 0 , some dangerous xy -**trace** is a prefix of a **trace** that leads ... 3 **MODEL CHECKER** IMPLEMENTATION ... change the seman- tics, that is, p 1 ::: pm are **disjoint** (if the ...
Cited by 7 - Web Search

[PS] PV: a **model-checker** for verifying ltl-x properties - group of 3 »
R Nalumasu, G Gopalakrishnan - Fourth NASA Langley Formal Methods Workshop, 1997 - techreports.larc.nasa.gov
Page 1. PV: A **Model-Checker** for Verifying LTL-X Properties Ratan Nalumasu Ganesh Gopalakrishnan ... that the domains of i ! I and i ! g are **disjoint** and that ...
Cited by 5 - View as HTML - Web Search - BL Direct

Designing a LTL **Model-Checker** Based on Unfolding Graphs - group of 2 »
JM Couvreur, S Grivet, D Poitrenaud - LECTURE NOTES IN COMPUTER SCIENCE, 2000 - Springer
... Designing a LTL **Model-Checker** Based on Unfolding Graphs ... We call **trace** function Φ AP the mapping in S → [AP ... Post is aP/T net (P and T are **disjoint** sets of ...
Cited by 3 - Web Search - BL Direct

Generating Test Oracles via Model Checking - group of 4 »
JR Callahan, SM Easterbrook, TL Montgomery - NASA/WVU Software Research Lab, Fairmont, WV, Technical ..., 1998 - cis.upenn.edu
... generation mechanism found in most **model checker** tools. ... into a complete cover of

disjoint equivalence partitions on ... If an inconsistency between a **trace** and the ...
Cited by 7 - View as HTML - Web Search

[PS] Mocha: Exploiting Modularity in Model Checking - group of 7 »
L de Alfaro, R Alur, R Grosu, T Henzinger, M Kang, ... - University of California at Berkeley Department of ..., 2000 - www-cad.eecs.berkeley.edu
... The symbolic **model checker** is based on BDD engines ... Implementation verication by checking **trace** containment between ... modules, if they have **disjoint** sets of ...
Cited by 3 - View as HTML - Web Search

Gooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

"model checker" disjoint trace Search

Google Home - About Google - About Google Scholar

Scholar

Results **1 - 10** of about **105** for **"model checker" mutually disjoint**. (**0.11** seconds)

All articles Recent articles

... Verification of a Behavioural Subset of UML Statechart Diagrams Using the SPIN **Model-checker** - group of 6 »
DA Latella, IA Majzik, MA Massink - Formal Aspects of Computing, 1999 - Springer
... as the work constitutes a basis for a PROMELA/SPIN based **model-checker** for UML ... where F is a finite set of sequential automata with **mutually disjoint** sets of ...
Cited by 110 - Web Search - BL Direct

Model Checking Coloured Petri Nets Exploiting Strongly Connected Components - group of 3 »
A Cheng, S Christensen, KH Mortensen - Proceedings of the International Workshop on Discrete Event ..., 1996 - daimi.au.dk
... expressing liveness properties since liveness is expressed by means of transition information.) For this purpose we introduce two **mutually** recursively defined ...
Cited by 27 - View as HTML - Web Search

Model Checking Complete Requirements Specifications Using Abstraction - group of 10 »
RV Bharadwaj, CLV Heitmeyer - Automated Software Engineering, 1999 - Springer
... Before practical software specifications can be analyzed efficiently using a **model checker**, the state explosion problem must be addressed, ie, the size of the ...
Cited by 90 - Web Search - BL Direct

Checking general safety criteria on UML statecharts - group of 5 »
Z Pap, I Majzik, A Pataricza - SAFECOMP, 2001 - Springer
... criterion: for each state and each trigger event, the guard conditions must be **mutually disjoint**. ... We use the **model checker** SPIN [10] as external tool to decide ...
Cited by 13 - Web Search - BL Direct

An Outline of PVS Semantics for UML Statecharts - group of 2 »
I Traore - Journal of Universal Computer Science, 2000 - jucs.org
... used to conduct a formal analysis using the PVS **modelchecker**. ... two or more concurrent substates or into **mutually** exclusive **disjoint** substates (also ...
Cited by 29 - Web Search

Model Checking and Other Ways of Automating Formal Methods - group of 6 »
J Rushby - Position paper for panel on Model Checking for Concurrent ..., 1995 - csl.sri.com
... a table construct, and can generate the proof obligations for **mutually disjoint** and exhaustive ... Using the **model checker** we are now also able to check certain ...
Cited by 5 - View as HTML - Web Search

Model-Checking Over Multi-Valued Logics - group of 5 »
M Chechik, S Easterbrook, V Petrovykh - Proceedings of FME'01, 2001 - Springer
... Given a system and a property, a **model checker** builds the reachability graph (explicitly or symbolically) by exhaustively exploring the state-space of the ...
Cited by 40 - Web Search - BL Direct

Experiences with the Application of Symbolic Model Checking to the Analysis of Software ... - group of 2 »
R Anderson, P Beame, W Chan, D Notkin - Lecture Notes in Computer Science, 1999 - Springer
... in terms of cases, it is natural require that the cases are **mutually disjoint**. ... were close to the maximum size which could be evaluated with a **model checker**. ...
Cited by 3 - Web Search - BL Direct

Applying the SCR Requirements Method to a Simple Autopilot - group of 9 »
R Bharadwaj, C Heitmeyer - Proceedings of the Fourth NASA Langley Formal Methods ..., 1997 - chacs.itd.nrl.navy.mil
... Forexample, Atlee and Gannon use a language based on logic to model the required behavior of a cruise control sys- tem 3 and a **model checker** to detect ...
Cited by 13 - View as HTML - Web Search - BL Direct

Efficient Decompositional Model Checking for Regular Timing Diagrams - group of 9 »
N Amla, EA Emerson, KS Namjoshi - Correct Hardware Design and Verification Methods: 10th IFIP ..., 1999 - Springer
... Sec- tion 4 describes how the algorithms are used with with the **model checker** VIS
[3 ... k and [k, k] as = k. – CD is a collection of **mutually disjoint** sets of ...
Cited by 16 - Web Search - BL Direct

"model checker" mutually disjoint Search